

Cloud based multicasting using fat tree data confidential recurrent neural network

P. Rajendiran, M. P. Karthikeyan, R. Venkatesan & B. Srinivasan

To cite this article: P. Rajendiran, M. P. Karthikeyan, R. Venkatesan & B. Srinivasan (2019) Cloud based multicasting using fat tree data confidential recurrent neural network, *Automatika*, 60:3, 305-313, DOI: [10.1080/00051144.2019.1597565](https://doi.org/10.1080/00051144.2019.1597565)

To link to this article: <https://doi.org/10.1080/00051144.2019.1597565>



© 2019 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group



Published online: 13 Jun 2019.



Submit your article to this journal [↗](#)



Article views: 271



View related articles [↗](#)



View Crossmark data [↗](#)



Cloud based multicasting using fat tree data confidential recurrent neural network

P. Rajendiran , M. P. Karthikeyan , R. Venkatesan and B. Srinivasan

School of Computing, SASTRA Deemed to be University, Thanjavur, India

ABSTRACT

With the progress of cloud computing, more users are attracted by its strong and cost-effective computation potentiality. Nevertheless, whether Cloud Service Providers can efficiently protect Cloud Users data confidentiality (DC) remains a demanding issue. The CU may execute several applications with multicast needs. In Cloud different techniques were used to provide DC with multicast necessities. In this work, we aim at ensuring DC in the cloud. This is achieved using a two-step technique, called Fat Tree Data Confidential Recurrent Neural Network (FT-DCRNN) in a cloud environment. The first step performs the construction of Fat Tree based on Multicast model. The aim to use Fat Tree with Multicast model is that the multicast model propagates traffic on multiple links. With the Degree Restrict Multicast Fat Tree construction algorithm using a reference function, the minimum average between two links is measured. With these measured links, multicast is said to be performed that in turn improves the throughput and efficiency of cloud service. Then, with the objective of providing DC for the multi-casted data or messages, DCRNN model is applied. With the Non-linear Recurrent Neural Network using Logistic Activation Function, by handling complex non-linear relationships, average response time is said to be reduced.

ARTICLE HISTORY

Received 3 January 2019
Accepted 16 March 2019

KEYWORDS

Cloud service providers;
cloud users; Fat Tree;
multicast; data confidential;
recurrent neural network

1. Introduction

Current progresses in data-as-a-service (DaaS) and cloud computing maintain data from numerous sources into tremendous repositories for queries. However, CSP is frequently a third-party delegate of the data owner, results of query related to integrity is not guaranteed and hence indispensable be authenticated. Privacy-Preserving Authentication framework for Aggregate (PAPA) [1] queries intended authenticated aggregate queries. PAPA affords integrity of query results and preserves the DC. It comprises authenticated index Merkle Grid tree (MG-tree) which has primitive multiset operations and the combination of aggregate query processing algorithms. Thus, it lessens computation and communication overheads for server and client. However, the current cloud environment constructs virtual overlay network with the objective of enabling the communication amongst several virtual machines that in turn connects the physical machines wasting network resources and creating bottlenecks, compromising the throughput and efficiency of a cloud service. A potential solution is to design the Multicast model using Fat Tree structure. The Fat Tree structure for the multicast model has very good potential at propagating traffic on multiple links.

Another main topic receiving remarkable attention is Network Virtualization. To this extent, multicast in

a cloud computing environment has become a distinguished research area, with precise observation to resource allocation issue of Multicast Virtual Networking (MVNs). Multicast Virtual Networking Embedding (MVNE) problem in [2] formulated Integer Linear Programming (ILP) with the objective to balance the load in the network. The MVNE first formed ILP.

With this, a 3-step approach, pruning the network, finding sub-graphs with potential feasible solutions and node mapping was performed. Finally, a Dynamic Programming (DP) approach was proposed to provide an optimal solution in polynomial-time with resource requirements in a homogeneous manner. However, with complex non-linear relationships, the average response time gets increased with the improper balancing of a load in the network.

A potential solution is to introduce Deep Neural Network (DNN) with which, a machine learning method, as opposed to task-oriented algorithms model complex non-linear relationships. Measures were taken to ensure DC for sensing big data streams using Selective Encryption [3] method. Despite DC, multicasting was not ensured. In [4], Cloud-based Data Centers were exploited to address multicasting, for minimizing the mapping cost.

Recently, several DC techniques have been designed in a cloud computing environment. However,

communication between several virtual machines and non-linear relationships requests remains a major issue to be addressed. Therefore, we present a novel DC technique. Initially, the virtual overlay network that in turn connects physical machines wasting the network resources is addressed by introducing a Fat Tree Data Structure that propagates traffic on multiple links, therefore improving the throughput of the cloud service.

Next, the design of multicast model with Fat Tree Data Structure using Degree Restrict Multicast Fat Tree construction algorithm improves the efficiency of the cloud service. Then, to attain DC, Data Confidential Recurrent Neural Network model is used. Here, by only creating the simulated packet for each CU, DC is said to be attained. Finally, using Logistic Activation Function, average response time is said to be further improved.

This paper is ordered as follows. Section 2 describes the related work. Section 3 describes the DC technique. Section 4 formally analyzes the experiments and focuses on simulation and evaluation. Finally, section 5 concludes the work.

2. Related works

In the past few years, with the swift deployment of bandwidth sensitive cloud-based applications, the demand for traffic demand in Internet data centre networks has been increasing exponentially. Hence, it necessitates the requirements for multicasting. In [5], adaptive modulation selection was applied with a fixed approximation ratio to improve the quality of transmission in cloud-based applications. However, the mapping cost remained unaddressed. To solve this issue, Survivable Multicast Service Oriented Virtual Network Mapping (SMVNM) [6] was examined to minimize the mapping cost and mapping time. Despite minimization involved in mapping, the DC was not focused. Attribute-based Encryption and Decryption algorithm were designed in [7] based on hierarchical authorization structure, ensuring scalability and fine-grained access control.

With the increase in the availability of diverse remote sensors, it has now become possible to acquire a diversified variety of information from several materials on the Earth. To name some of them are information regarding spectral acquired through passive sensors, acquiring height and information regarding shape through a light detection and ranging (LiDAR) sensors for a cloud. In [8], Convolutional Neural Network with Deep Learning was investigated for accurate classification for cloud analysis. Yet another DNN using Hierarchical Fused Fuzzy component was presented in [9] for the data to be classified.

Ensuring reliability support for services hosted in cloud computing environments is a well-studied area that has gained the attention of researchers from the

literature in recent years. However, the existing methods did not provide mechanisms for the communication mode that these hosted services may exhibit. In [10], the ILP model was designed with the objective of restoring failed services in terms of restorable ratio. However, the failure prone nature of CSP poses threats. To address this issue, Optimal Polynomial Time algorithm was investigated in [11] for multicast services residing in cloud networks. Yet another multicast tree was designed in [12] for maintaining requested Quality of Service (QoS), ensuring restoration ratio and resulting in considerably fast execution time.

In recent years, the requirement of resources is increasing exponentially and hence processing and maintaining enormous data in several fields remains a major concern to be addressed. However, acquiring these resources and conserving them is considered to be a tremendous task. With Cloud computing, these resources are said to be obtained on demand with meagre management effort. However, ensuring confidentiality of data is considered to be promising challenge. In [13], homomorphic encryption was applied using RSA ensuring DC. Certain issues related to DC, privacy, and integrity were presented in [14].

Cloud computing environment lessens computational cost and overhead for CU. Though, supplementary technical safeguards are presented to secure and confidential for CU. In [15], cryptographic functionality was exploited for user privacy, data minimization and authentication of stored and processed data.

Asymmetric scalar-product-preserving encryption scheme was studied in [16], to control the privacy of data. Successful processing of data and analysis for Geological Information Services was addressed in [17] with the aid of aggregation. Challenges related to security and privacy was analyzed in [18]. However, with the increase in the search space, computation overhead also got increased. To address this issue, Oblivious Similarity-based Search was applied in [19] by only encrypting the relevant data and therefore avoiding unnecessary computation. As a result, computation overhead was said to be minimized by applying an encrypted bloom filter and probabilistic homomorphic encryption. Yet another method based on parallelization was introduced in [20] to address the DC and computation overhead involved during the same process. This was said to be achieved using hypothesis model.

The telemetry system for the diagnosis of Asthma and Chronic obstructive pulmonary disease (COPD) was presented in [21]. However, the time diagnosis of the disease was high. Recovery algorithm for wormhole and the isolation of the black hole was introduced in [22] for enhancing the performance of the network. However, the attack detection rate was not improved. Secure framework for storing data on the cloud was

developed based on an encryption scheme in [23]. But, the throughput was remained unaddressed.

Nevertheless, the flexibility of DC has not been fully explored for multicast provisioning. Therefore, in this work, DC is addressed by modifying the existing multicast virtual network in a cloud computing environment with the aid FT-DCRNN technique.

3. Methodology

In this section, a new technique named FT-DCRNN which is used to increase the DC by modifying the existing multicast virtual network method for efficient multicast in a cloud computing environment is presented. The proposed algorithm provides several advantages such as average response time, throughput, efficiency and adaptability during multicasting for cloud provisioning.

Here, intelligent nodes, called, primary cloud users, a relay node and edge nodes are used for constructing Fat Tree based on the average delay and to find the multicast model with low cost. Figure 1 given below shows the block diagram of FT-DCRNN technique.

Next, with the constructed Fat Tree, hidden layer vector and output vector with the activation function separately are identified to ensure DC. It is executed by discovers activation function at different time intervals, improving DC. FT-DCRNN technique implemented by Cloudsim simulation and confirms that the proposed algorithm affords better performance than the other existing algorithms.

3.1. Fat tree data structure

There are numerous types of content to be distributed in a cloud computing environment, such as newsfeed, updates related to social, weather forecasting information and so on. The CSPs distribute the content to only an insignificant fragment of cloud users. Then, this CU walk around and communicate other CU strategically and interchange the information through several means.

As far as throughput and efficiency of the cloud service are concerned, designing an efficient data confidential method that delivers the information to the intended CU becomes a multicast problem in a cloud computing environment. In a cloud computing environment, a relay node " $RN = \{rn_1, rn_2, \dots, rn_n\}$ " is an encountered node that store-carry-and-forward the message or information to the cloud servers (CS) or destinations swiftly with a small overhead. Therefore, designing a satisfactory formula or benchmark to identify good relay nodes becomes a demanding issue in cloud service multicast provisioning. This is performed in the proposed method using a Fat Tree data structure.

In a Fat Tree https://en.wikipedia.org/wiki/Data_structure data structure, every branch has the same

thickness or breadth, despite of its positioning in the hierarchy. In other words, every branch is said to be skinny, i.e. low-bandwidth. In a Fat Tree https://en.wikipedia.org/wiki/Data_structure data structure, branches that are neighbouring the top of the hierarchy are thicker than branches further down the hierarchy. In the cloud environment, the branches are otherwise referred to as data links and the heterogeneous thickness or breadth (i.e. bandwidth) of the data links allows for ensuring DC. This is attained by enabling communication between several virtual machines, therefore improving the throughput and efficiency of the cloud service. Figure 2 shows the Fat Tree data structure.

As illustrated in Figure 1, the Fat Tree Data Structure includes a three-degree tier, involving primary cloud user "CU", relay node "RN" and edge node "EN" respectively. The three-degree tier has a primary node or primary CU in the root of the tree, an assembling node or relay node (i.e. relay cloud user) in the middle and an edge node at the third level of the fat tree. In this work, the multicast problem in a cloud computing environment using Fat Tree Data Structure is considered.

Multicast is a service where a cloud user $CU = \{cu_1, cu_2, \dots, cu_n\}$ " $CU = \{cu_1, cu_2, \dots, cu_n\}$ " sends the message or information " $M = m_1, m_2, \dots, m_n$ " to all of the cloud service providers " $CSP = \{csp_1, csp_2, \dots, csp_n\}$ " or the destinations. In the FT-DCRNN technique, the problem of multicast is solved by proposing a Degree Restrict Multicast Fat Tree construction algorithm. The pseudo-code representation of Degree Restrict Multicast Fat Tree construction algorithm is given below.

Algorithm 1. Degree Restrict Multicast Fat Tree construction algorithm

Input: cloud user ' $CU = \{cu_1, cu_2, \dots, cu_n\}$ ', message ' $M = m_1, m_2, \dots, m_n$ ', cloud service providers ' $CSP = \{csp_1, csp_2, \dots, csp_n\}$ ', relay node ' $RN = \{rn_1, rn_2, \dots, rn_n\}$ ' edge node ' $EN = \{en_1, en_2, \dots, en_n\}$ '

Output: throughput and delay efficient fat tree construction

```

1:Begin
2: For each cloud user 'CU' with 'CSP' and message 'M' to be sent
3:   Measure average delay 'AvgDelay' using equation (1)
4:   Measure the number of relay nodes using equation (3) and (4)
5:   Measure total number of data links using equation (5)
6:   If message 'M' arrived then
7:     If data link ∈ SET 'EN – RN' THEN
8:       Let 'DN – > Destinationnode'
9:       Let 'SN – > Sourcenode'
10:      Let next node = 'Lookup(DN, SN)' using equation (3)
11:     Else
12:       If link ∈ SET 'RN – CU' Then
13:         Let 'NextDataLink' = unused available link
14:         Let next node = 'Lookup(NextDataLink)' using equation (3)
15:       Else
16:         Let 'DN – > Destinationnode'
17:         Let 'NextDataLink = Lookup(DN)' using equation (3)
18:       End if
19:     End if
20:   End if
21: End for
22: End

```

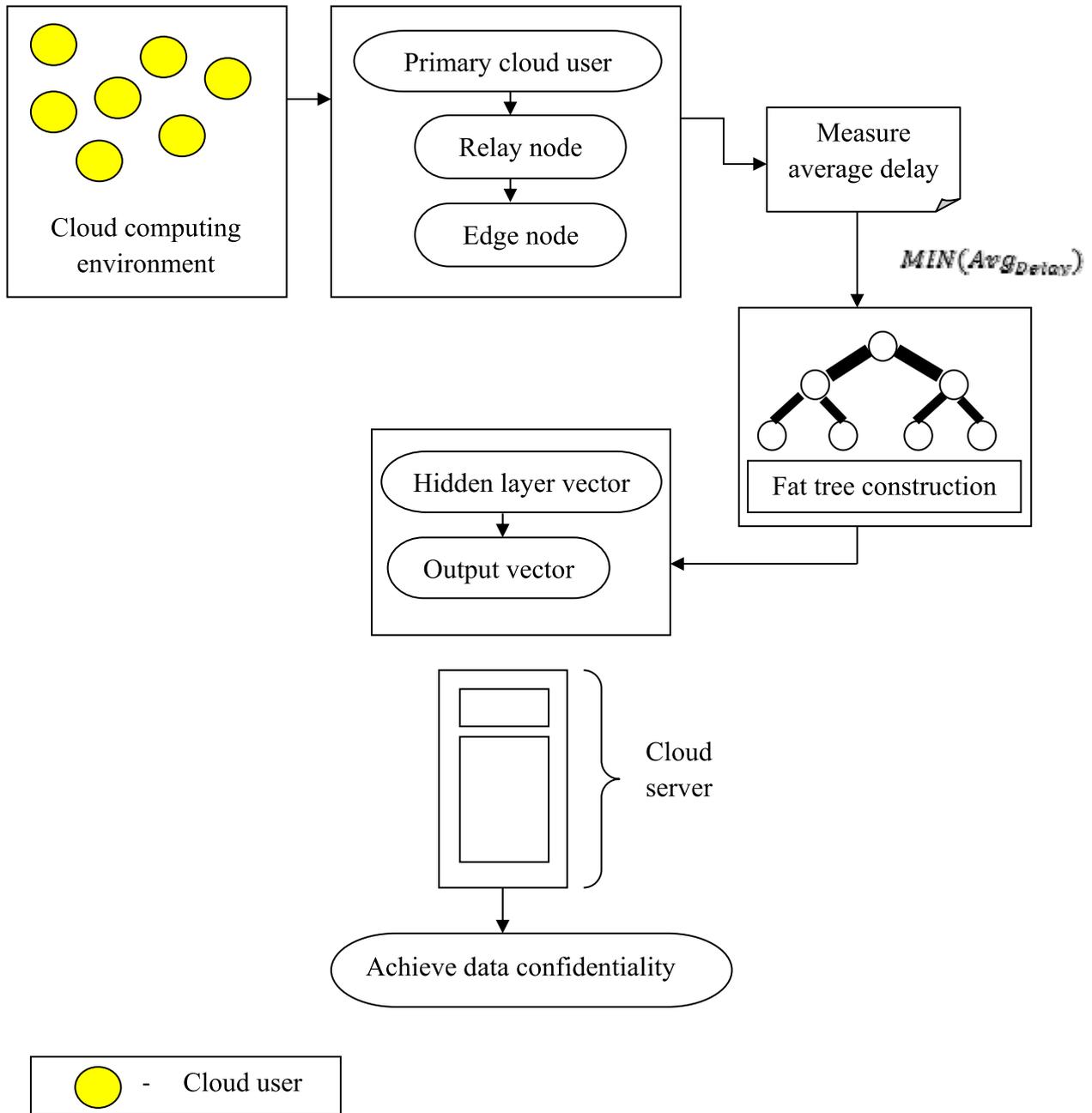


Figure 1. Block diagram of fat tree data confidential recurrent neural network.

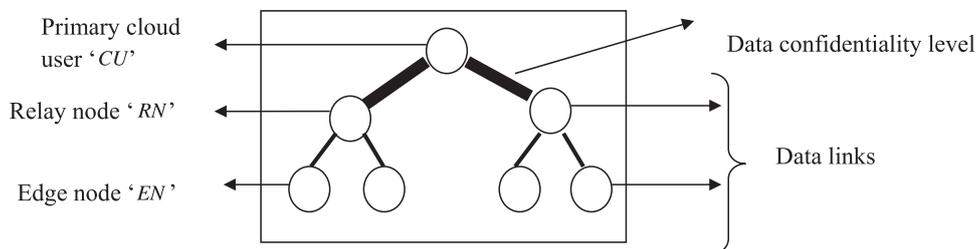


Figure 2. Fat tree data structure.

As given in the above algorithm 1, in this Fat Tree data structure, that there is no single bottleneck for replication and traffic transmission while using a relay node. In addition, the Fat Tree data structure reduces the traffic overhead on the cloud network, and control the maximum and average delay, therefore increasing

the throughput and efficiency of cloud service. Let us measure the average delay during multicast between “(i, j)” and “(m, n)”. It is mathematically formulated as given below.

$$AvgDelay(i, j) = RI(R_i^j, R_i^{j+1}) * R_a + Dis(R_i^j, R_i^{j+1}) * R_t \tag{1}$$

$$\begin{aligned} Avg_{Delay}(m, n) = & RI(R_m^n, R_m^{n+1}) * R_a \\ & + Dis(R_m^n, R_m^{n+1}) * R_t \end{aligned} \quad (2)$$

From the above equations (1 and 2), the average delay “ Avg_{Delay} ” is measured using the relay index “ $RI(R_i^j, R_i^{j+1})$ ”, the “ j th” index node on the path from the CU “i.e. source” to CS. Now let us rewrite “ D_i^a ” and “ D_i^t ” as follows.

$$D_i^a = RI(R_i^j, R_i^{j+1}) \quad (3)$$

$$D_i^t = Dis(R_i^j, R_i^{j+1}) \quad (4)$$

From the above equations (3 and 4), a number of relay nodes during the entire period of transmitting a packet or message or information from the CU to the CS is given by “ D_i^a ”. On the other hand, “ D_i^t ” symbolizes the total number of links during the entire period of transmitting a packet or message or information from the CU to the CS. Then, the above equations (1 and 2) is rewritten as follows.

$$Avg_{Delay} = D_i^a * R_a + D_i^t * R_t \quad (5)$$

The function “ $Lookup$ ”, that occurs in step 10, step 14 and step 17, is most important in this algorithm. It is a reference function used to assign the minimum of average delay between to data links. It is mathematically formulated as given below

$$Lookup = MIN(Avg_{Delay}(i, j), Avg_{Delay}(m, n)) \quad (6)$$

From the above equation (6), the result of the lookup function remains the minimum of the average delay between two data links. The minimum average delay incurred is considered as the final data link through which multicast is said to be performed. In this way, the throughput and efficiency of the cloud service are said to be improved.

3.2. Data confidential recurrent neural network model

Preserving DC is a critical issue when disseminating microdata related geographical oceanographic and surface meteorological readings for public-use in a cloud computing environment. Despite throughput and efficiency being achieved using Fat Tree data structure, preserving DC, with complex non-linear relationships has to be addressed.

In this work, the problem is approached by using Recurrent Neural Network model that produces simulated data closely related to raw data as given by CU but different for each item. As only simulated data are produced for the closely related raw data, both average response time and DC rate is said to be improved. Figure 3 shows the Data Confidential Recurrent Neural Network model.

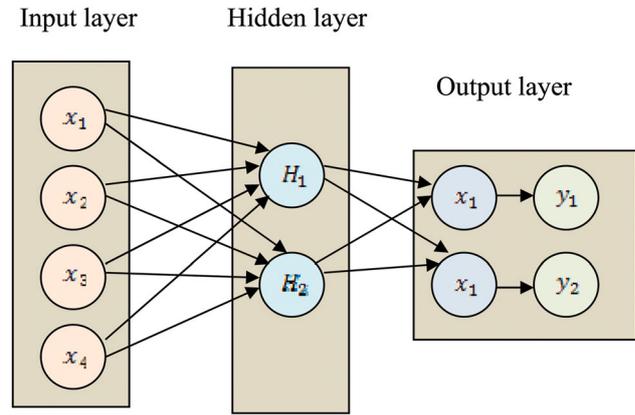


Figure 3. Structure of recurrent neural network model.

The above figure involves the input layer, hidden layer, and output layer. From the above figure, for four input layer “ x_1, x_2, x_3, x_4 ”, with two hidden layers “ H_1 and H_2 ” exploit complex non-linear relationships resulting in the output layer “ y_1, y_2 ”. The pseudo-code representation of Data Confidential Recurrent Neural Network is as given below.

Algorithm 2. Data Confidential Recurrent Neural Network algorithm

Input: Input: cloud user ‘ $CU = \{cu_1, cu_2, \dots, cu_n\}$ ’, message ‘ $M = m_1, m_2, \dots, m_n$ ’, cloud service providers ‘ $CSP = \{csp_1, csp_2, \dots, csp_n\}$ ’, ‘ $RN = \{rn_1, rn_2, \dots, rn_n\}$ ’, time series ‘ t ’, weight matrices ‘ W_{ih} ’, ‘ W_{oh} ’ and ‘ W_{hh} ’,

Output: Data confidential multicast message transmission

- 1: **Begin**
- 2: **For** each cloud user ‘ CU ’ with ‘ CSP ’ and confidential data or message ‘ M ’ to be sent
- 3: Measure hidden layer vector using equation (7)
- 4: Measure output vector using equation (8)
- 5: Measure activation function for hidden layer vector using equation (9)
- 6: Measure activation function for output vector using equation (10)
- 7: **End for**
- 8: **End**

As given in the above algorithm, let “ $x(t)$ ” and “ $y(t)$ ” represents the input layers for time series “ t ” respectively with weight matrices representing “ W_{ih} ”, “ W_{oh} ” and “ W_{hh} ”. Here, “ W_{ih} ” corresponds to the weight between the input and hidden layers, “ W_{oh} ” corresponds to the weight between output and hidden and “ W_{hh} ” corresponds to the weight between two hidden layers. The hidden layer vector and output vector is symbolized as below.

$$H_t = AF_H(W_H x_t + Y_H H_t + b_H) \quad (7)$$

$$Y_t = AF_Y(W_Y H_t + b_Y) \quad (8)$$

From equations (7 and 8), hidden layer vector “ H_t ” and output vector “ Y_t ” are acquired from the activation function of hidden vector “ AF_H ” and output vector “ AF_Y ” respectively. To this, also, the hidden vector “ H_t ” and input vector “ x_t ” are also considered, where

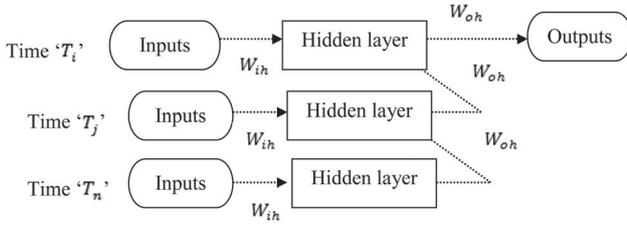


Figure 4. Structural representation of non-linear recurrent neural network.

“ W ”, “ Y ” and “ b ” corresponds to the parameter indices. Figure 4 shows the non-linear recurrent neural network structures at different time intervals.

As shown in the above figure, Non-linear Recurrent Neural Network is employed to send the data to the cloud users with higher confidentiality. The input layer consists of time series with a weighted matrix. The initial or first layer has a weight which acquired from the input layer, each layer acquiring the weight from the previous layer. Logistic Activation Function in the hidden layer and the output layer is used to safeguard CU data with the objective of reducing average response time. It is mathematically represented as given below.

$$AF_H(t) = H * \left[\frac{1}{1 + e^{-t}} \right] \quad (9)$$

$$AF_Y(t) = Y * \left[\frac{1}{1 + e^{-t}} \right] \quad (10)$$

From the above equations (9 and 10), with separate activation function for both the hidden layer vector and output layers at different time intervals and also using simulated data, DC is said to be achieved.

4. Simulation and performance analysis

This section analyzes the impact of the proposed FT-DCRNN technique via Cloud Sim simulator. The cloudsim goal is to provide a global and extensible simulation framework that compares the proposed FT-DCRNN technique with the existing PAPA [1] and MVNE [2]. The experiment is conducted to measure and evaluate the FT-DCRNN technique on the factors such as throughput, efficiency, average response time and DC.

Dataset exploited is ENSO dataset. ENSO dataset comprises Wind, Oscillation Index, Sea Surface Temperature and Outgoing Long Wave Radiation. The dataset comprises oceanographic and surface meteorological readings acquired from series of buoys located in equatorial Pacific. With this dataset, CU spread throughout the world interacts with the other CU through CSP in predicting weather conditions, variables possessing greater effect on the climate variations and so on.

4.1. Performance evaluation

Performance of FT-DCRNN technique is evaluated in this section and compared with state-of-the-art methods.

4.2. Impact of throughput and efficiency

Throughput and efficiency of FT-DCRNN technique is evaluated with PAPA [1] and MVNE [2] using the number of user requests as input in the cloud environment in the comparison experiments. The two imperative factors are considered for evaluating infrastructure services afforded by the cloud environment are Throughput and Efficiency. Throughput computes number of CU requests or number of tasks finished by CSP at a particular interval of time. Several factors influence the rate of throughput and hence resulting in a positive or negative impact on the task execution.

Let us consider a scenario with CU possessing “ i ” tasks or user requests and it submitted to execute on “ j ” machines from CSP. Consider “ Exe_{time} ” corresponds to the execution time of “ i ” tasks on “ j ” machines and let “ OH_{time} ” symbolizes overhead time, owing to numerous factors like cloud service environment initiation delays and inter-task communication delays. Thus throughput of Cloud service is given below.

$$T = \frac{i}{Exe_{time}(i,j) + OH_{time}} \quad (11)$$

From the above equation (11), the throughput value “ T ” is measured based on the execution time “ Exe_{time} ” and the overhead time “ OH_{time} ”. Table 1 given below provides the throughput and efficiency distribution of all the three techniques, FT-DCRNN, PAPA [1] and MVNE [2]. This experiment is set to verify the high effectiveness of the throughput against the CU requests placed in a cloud environment. In this experiment environment, the Fat Tree Data Structure is combined with the Multicast model to achieve the high effectiveness of the throughput. The throughput is compared with PAPA [1] and MVNE [2] by capturing the same amount of CU tasks.

Table 1 given above provides the comparison of throughput for diverse number of CU requests in setup phase, from which it is inferred that: 1) the throughput is proportional to the number of CU requests, and 2) to handle the same number of CU requests, FT-DCRNN technique achieves higher amount of throughput than PAPA and MVNE. From the table, rate of throughput increases with the increase in number of CU requests and includes insignificant gaps when the number of CU requests is greater than 40. That is to say, the rate of throughput in FT-DCRNN technique is higher than those in PAPA and MVNE. This is because of the application of Fat Tree Data Structure in the FT-DCRNN technique. Motivations for which is twofold.

Table 1. Comparison of throughput.

Cloud user requests	Throughput (%)			Efficiency (%)		
	FT-DCRNN	PAPA	MVNE	FT-DCRNN	PAPA	MVNE
10	73.84	69.25	67.13	78.32	75.56	74.23
20	74.95	71.53	70.13	80.14	78.23	75.56
30	78.22	76.53	75.13	81.32	79.11	77.32
40	81.53	78.31	76.53	84.12	80.18	79.23
50	82.67	80.14	78.12	85.56	82.35	80.98
60	83.32	81.56	80.29	87.11	84.14	82.13
70	84.13	82.35	81.12	89.23	86.67	84.33
80	85.68	83.12	81.99	90.43	88.12	85.65
90	87.92	85.56	83.12	90.89	89.23	86.76
100	90.24	87.69	84.56	92.14	90.05	87.78

Initially, it stores each data in CS using linear programming methods over various substrate networks. On the other hand in FT-DCRNN technique, a cloud service multicast provisioning was applied to measure the relay index. The relay index was measured using Degree Restrict Multicast Fat Tree construction algorithm depending on the agreement between CU and CSP.

Second, for performing multicast, existing PAPA and MVNE techniques used virtual overlay network for communication between CU and CSPs, which is said to be affected in the case of the huge traffic. On the other hand, in FT-DCRNN technique, multicast was first said to be performed based on the average delay. Then, with the minimum average delay, and then the Fat Tree Data Structure was constructed. This, in turn, improved the throughput using optimization for storage was first said to be attained using FT-DCRNN technique by 3% compared to PAPA and 6% compared to MVNE.

Next, the cloud system efficiency of FT-DCRNN technique is evaluated and compared them with PAPA and MVNE based on a number of CU requests in a cloud environment. Cloud system efficiency indicates the capable use of lease services. Hence, higher efficiency denotes lower overhead. System efficiency is mathematically given as below.

$$SE = \frac{Exe_{time}(i,j)}{Exe_{time}(i,j) + OH_{time}} \quad (12)$$

From above equation (12), while measuring the cloud system efficiency “SE” both, execution time “ Exe_{time} ” and overhead time “ OH_{time} ” are considered. Cloud system efficiency during multicast is one of the challenges to be addressed during rendering cloud service. The comparison of system efficiency for FT-DCRNN technique is measured and compared with PAPA and MVNE is listed in Table 1. The results reported in Table 1, prove that with enhance in the number of CU request, the efficiency is enhanced.

As provided in Table 1, the FT-DCRNN technique performs relatively well when compared to two other techniques PAPA and MVNE. The system efficiency using FT-DCRNN technique is improved by comparing the requirements of the CU requests with that of the

data link based on the relay node and edge node. The advantage of arriving at data link using Degree Restrict Multicast Fat Tree construction algorithm remains in deriving the next node or next data link, that avoids the single CU request being held up by the CS, given scope for other CU requests. So, every user is ensured with message transmission and therefore improving the overall efficiency of the cloud using FT-DCRNN technique by 3% compared to PAPA and 6% compared to MVNE.

4.3. Impact of average response time

The efficiency of cloud service availability is evaluated with average response time. It is evaluated as given below.

$$AvgRes_{time} = \sum_{i=1,2,\dots,n} \frac{Time_i}{n} \quad (13)$$

From above equation (13), the average response time “ $AvgRes_{time}$ ” is measured with the ratio of time between cloud users “ i ” requested for a service or sent a message to another CU and is actually accessible. The total number of service requests are symbolized by “ n ”. It is evaluated in milliseconds (ms).

Figure 5 portrays the impact of average response time versus a different number of CU in the range of 20–200 with three techniques. The results of 10 different CU are provided in the above figure. While the number of CU was 120, FT-DCRNN technique consumes 0.57 ms whereas the PAPA [1] and MVNE

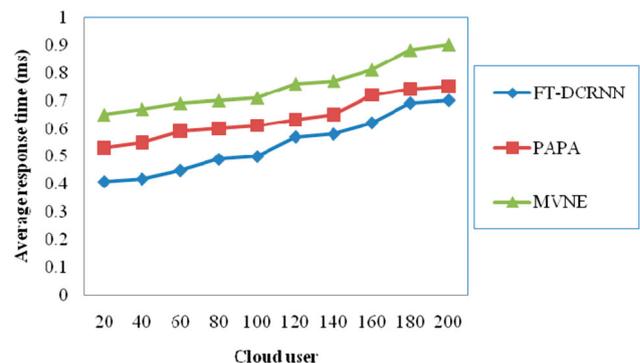
**Figure 5.** Graphical representation of average response time.

Table 2. Tabulation for data confidentiality observed using FT-DCRNN, PAPA and MVNE.

Techniques	Data Confidentiality (%)									
	Cloud Users									
	20	40	60	80	100	120	140	160	180	200
FT-DCRNN	68.35	71.45	72.59	75.69	70.28	74.89	78.92	80.14	84.35	86.24
PAPA	64.13	67.25	68.15	71.35	75.89	72.13	77.98	79.13	81.35	82.45
MVNE	63.45	65.13	68.98	70.22	68.91	73.24	75.67	77.25	80.45	81.23

[2] consume 0.63 ms and 0.76 ms respectively. Therefore, the average response time using the proposed FT-DCRNN technique is lower as compared to other exiting techniques.

Besides, by increasing the number of cloud users, data to be allocated to the CU gets increased and therefore, the average response time also gets increased. Hence, the number of CU is inversely proportional to the average response time. This is because of the application of Data Confidential Recurrent Neural Network model. By applying Data Confidential Recurrent Neural Network model, the CSP presents only the simulated data closely interconnected to the input raw data as given by cloud user. But the simulated data produced by the CSP is different for different cloud users. With this generated simulated data average response time using FT-DCRNN technique is reduced by 15% compared to PAPA and 29% compared to MVNE.

4.4. Impact of data confidentiality

DC in the cloud computing environment or cloud service provisioning refers to protecting information or message from being accessed by unauthorized cloud users. In other words, only the CU who is authorized to do so can gain access to sensitive information or message. Table 2 summarizes the DC achieved using FT-DCRNN technique, PAPA and MVNE respectively. In this table, DC achieved for the different number of CU ranges from 20 to 200.

As provided in the above table, the FT-DCRNN technique provides better DC level as compared to PAPA [1] and MVNE [2]. Besides, while increasing the number of cloud users, DC also gets increased, but the DC level fell down with 100 CU using FT-DCRNN, 120 CU using PAPA and 100 CU using MVNE. But comparative analysis shows that the DC using FT-DCRNN technique is improved as compared to other existing techniques.

This is owing to Data Confidential Recurrent Neural Network algorithm in FT-DCRNN technique where only simulated data are obtained by the CSP using Recurrent Neural Network method. Finally, with separate activation function for both the hidden layer vector and output layers at different time intervals, DC using FT-DCRNN technique is said to be improved by 7% when compared to PAPA and 8% when compared to MVNE respectively.

5. Conclusion

FT-DCRNN technique is designed for attaining DC in a cloud environment. This technique improves the throughput and efficiency by performing multicasting with the help of Fat Tree data structure. The technique uses Degree Restrict Multicast Fat Tree construction algorithm for optimal communication amongst several virtual machines. By applying the multicast model in FT-DCRNN technique, traffic on multiple links are readily propagated. Finally, by applying Recurrent Neural Network simulated data are produced, that is then sent to the cloud users, ensuring DC and minimizing average response time. A series of experiments are conducted to test the throughput, efficiency of cloud service, average response time and DC. The throughput and efficiency of cloud service are improved to 9% with the average response time being reduced to 29% when compared with the state-of-the-art methods.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

P. Rajendiran  <http://orcid.org/0000-0001-9860-5072>

M.P. Karthikeyan  <http://orcid.org/0000-0001-9989-9041>

R. Venkatesan  <http://orcid.org/0000-0003-3057-6580>

B. Srinivasan  <http://orcid.org/0000-0002-6877-3435>

References

- [1] Xu C, Chen Q, Hu H, et al. Authenticating aggregate queries over set-valued data with confidentiality. *IEEE T Knowl Data En.* 2017 Nov;(99):630–640.
- [2] Ayoubi S, Assi C, Shaban K, et al. MINTED: multicast VIRTUAL NeTwork Embedding in cloud data centers with delay constraints. *Ieee T Commun.* 2015 Apr;63(4):1291–1305.
- [3] Puthal D, Wu X, Nepal S, et al. SEEN: a selective encryption method to ensure confidentiality for big sensing data streams. *IEEE T Big Data.* 2017;(99):1–14.
- [4] Liao D, Sun G, Anand V, et al. Efficient provisioning for multicast virtual network under single regional failure in cloud-based datacenters. *KSII T Internet Inf.* 2014 Jul;8(7):2325–2349.
- [5] Zhu Z, Liu X, Wang Y, et al. Impairment- and splitting-aware cloud-ready multicast provisioning in elastic optical networks. *IEEE ACM T Network.* 2017 Apr;25(2):1220–1234.
- [6] Liao D, Sun G, Anand V, et al. Survivable provisioning for multicast service oriented virtual network requests

- in cloud-based data centers. *Opt Switch Netw.* Elsevier. 2014 Jun;14:260–273.
- [7] Teng W, Yang G, Xiang Y, et al. Attribute-based access control with constant-size ciphertext in cloud computing. *IEEE T Cloud.* 2015 Jun.
- [8] Ghamisi P, Höfle B, Zhu XX. Hyperspectral and lidar data fusion using extinction profiles and deep convolutional neural network. *IEEE J-Stars.* 2017 Jun;10(6):3011–3024.
- [9] Deng Y, Ren Z, Kong Y, et al. A hierarchical fused fuzzy deep neural network for data classification. *IEEE T Fuzzy Syst.* 2017 Aug;25(4):1006–1012.
- [10] Ayoubia S, Assia C, Chena Y, et al. Restoration methods for cloud multicast virtual networks. *J Netw Comput Appl.* Elsevier. 2017 Mar;78:180–190.
- [11] Ayoubi S, Assi C, Narayanan L, et al. Optimal polynomial time algorithm for restoring multicast cloud services. *IEEE Commun Lett.* 2016 Aug;20(8):1543–1546.
- [12] Ayoubi S, Chen Y, Assi C, et al. Multicast tree repair and maintenance in the cloud. *IEEE 8th International Conference on Cloud Computing*; 2015 Jun.
- [13] Suveetha K, Manju T. Ensuring confidentiality of cloud data using homomorphic encryption. *Indian J SCI Technol.* 2016 Feb;9(8). doi:10.17485/ijst/2016/v9i8/87964.
- [14] Aldossary S, Allen W. Data security, privacy, availability and integrity in cloud computing: issues and current solutions. (*IJACSA*) *Int J Adv Comput Sci Appl.* 2016;7(4):485–498.
- [15] Henrich TL, Ghernaouti S. Selected cloud security patterns to improve end user security and privacy in public clouds. *Private Technol Pol.* Springer. 2014 Jul;9857:115–132.
- [16] Hong J, Wen T, Guo Q, et al. Secure kNN computation and integrity assurance of data outsourcing in the cloud. *Math Probl Eng.* 2017;2017:1–15.
- [17] Zhu Y, Tan Y, Luo X, et al. Big data management for cloud-enabled geological information services.
- [18] Lu R, Rahulamathavan Y, Zhu H, et al. Security and privacy challenges in vehicular cloud computing. *Hindawi Publishing Mob Inf Syst.* 2016.
- [19] Pervez Z, Ahmad M, Khattak AM, et al. OS2: oblivious similarity based searching for encrypted data outsourced to an untrusted domain. *PLOS ONE.* 2017 Jul 10. doi:10.1371/journal.pone.0179720.
- [20] Alam M, Ihsan A, Khan MA, et al. Optimizing SIEM throughput on the cloud using parallelization. *PLOS ONE.* 2016 Nov 16. doi:10.1371/journal.pone.0162746.
- [21] Granulo E, Becar L, Gurbeta L, et al. Telemetry system for diagnosis of asthma and chronic obstructive pulmonary disease (COPD). *Internet of Things Technologies for HealthCare*, Mobyen Uddid Ahmed, Shahina Begum, Wasim Raad, Springer International Publishing; 2016. p. 113–118. doi:10.1007/978-3-319-51234-1_18.
- [22] Jyothirmai P, Raj JS, Smys S. Secured self organizing network architecture in wireless personal networks. *Wireless Pers Commun.* 2017;96(4):5603–5620.
- [23] Praveena A, Smys S. Ensuring data security in cloud based social networks. *International conference of Electronics, Communication and Aerospace Technology (ICECA)*, IEEE; 2017, vol. 2. p. 289–295.